Name: Sahib Bajwa

ID: 107553096

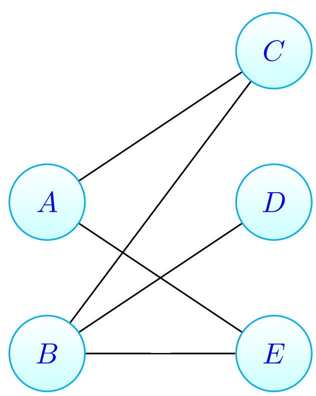
CSCI 3104, Algorithms **Profs. Chen & Grochow**

Quiz 8 Q1 S17 **Spring 2020, CU-Boulder**

Instructions: This quiz is open book and open note. You **may** post clarification questions to Piazza, with the understanding that you may not receive an answer in time and posting does count towards your time limit (30 min for 1x, 37.5 min for 1.5x, 45 min for 2x). Questions posted to Piazza **must be posted as PRIVATE QUESTIONS.** Other use of the internet, including searching for answers or posting to sites like Chegg, is strictly prohibited. Violations of these are grounds to receive a 0 on this quiz. Proofs should be written in **complete sentences. Show and justify all work to receive full credit.**

Standard 17. Let $G(V, E)$ be a graph. A *matching* of G is a set of edge $\mathcal{M}$ such that no two edges in $\mathcal{M}$ share a common vertex. That is, if $(i, j), (u, v) \in \mathcal{M}$ are distinct edges, then $i \neq u, i \neq v$, $j \neq u$, and $j \neq v$.

A graph is *bipartite* if its vertices can be partitioned into two sets $V(G) = L \cup R$ such that every edge has one endpoint in L and one endpoint in R. Note that L and R are disjoint. The graph pictured below is an example.



Consider the following problem

Bipartite Maximum Matching
Input: A bipartite graph $G = (L, R; E)$
Output: A matching $\mathcal{M} \subseteq E(G)$ whose size $|\mathcal{M}|$ is as large as possible.

Name: Sahib Bajwa
ID: 107553096

CSCI 3104, Algorithms **Profs. Chen & Grochow**
Quiz 8 Q1 S17 **Spring 2020, CU-Boulder**

(a) Describe how to reduce the above problem to the (one-source, one-sink) max-flow problem from class. Your description should be **general**, and not tied to a specific example. (You may illustrate with an example for expository purposes, but an example alone is not sufficient. E.g., "This is how my construction is performed in general. Then for example, this is how we apply the construction to the graph I selected.")

To reduce this problem to the max-flow problem from class, we create two new vertexes called s and t. The s vertex will have edges to every vertex in L and t will have edges to every vertex in R. Now, we have a one-source s, one-sink t max flow problem. Then for this example, A and B will be in L and connected to s by an edge each and C, D, and E will be connected to t through an edge each.

(b) Using your reduction, find a maximum matching in the graph above. Show your work, as well as your final answer. Note that there may be multiple maximum matchings in the graph above; you need only find one such matching.

From the reduction above, we can set each of the new edges we created to have a capacity of 1. Then, we can run the Ford-Fulkerson algorithm on the graph to find the maximum matching.
First, the algorithm will take the path s $\rightarrow$ A $\rightarrow$ C $\rightarrow$ t. Considering all of these edges have a capacity of 1, they are removed from the possibility of being used again to push flow.
Next, the algorithm will take the path s $\rightarrow$ B $\rightarrow$ D $\rightarrow$ t. Considering all of these edges also have a capacity of 1, theyu are removed from the possibility of being used again to push flow.
Since there is now no way to push more flow, the algorithm is finished. The max-flow of this graph is now the maximum matching of the graph as well.
(A, C), (B, D)